МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего профессионального образования

«Национальный исследовательский ядерный университет «МИФИ»

ФАКУЛЬТЕТ КИБЕРНЕТИКИ И ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

КАФЕДРА КОМПЬЮТЕРНЫХ СИСТЕМ И ТЕХНОЛОГИЙ (№ 12)

«УтверждЕН

на заседании кафедры

«\_\_\_»\_\_\_\_\_\_\_\_\_\_20\_\_ г.,

протокол №\_\_\_\_\_\_\_\_\_

Зав. каф. 12

\_\_\_\_\_\_\_\_\_\_\_\_\_/М. А. Иванов/

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ**

по дисциплине

**«НИЗКОУРОВНЕВОЕ ПРОГРАММИРОВАНИЕ»**

|  |  |
| --- | --- |
| Направление подготовки (специальность) | 230100.62 Информатика и вычислительная техника |
|  |  |
| Профиль подготовки (при его наличии) |  |
|  |  |
| Наименование образовательной программы (специализация) | Высокопроизводительные компьютерные системы и технологии |
|  |  |
| Квалификация (степень) выпускника | бакалавр |
|  |  |
| Форма обучения | очная |

г. Москва, 2015 г.

**ПАСПОРТ**

**фонда оценочных средств**

**по дисциплине «Низкоуровневое программирование»**

(наименование дисциплины)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Семестр** | **Интерактив** | **Трудоемкость, кред.** | **Общий объем курса, час.** | **Лекции, час.** | **Практич. занятия, час.** | **Лаборат. работы, час.** | **СРС, час.** | **КСР, час.** | **Форма(ы) контроля, экз./зач./КР/КП** |
| 4 | 0.00 | 2 | 72 | 30 | 0 | 15 | 27 | 0 | Зачет |
| ИТОГО | 0.00 | 2 | 72 | 30 | 0 | 15 | 27 | 0 |  |

Группы: К04-121, К04-122, К04-123, К04-12В, К04-12С

**Модели контролируемых компетенций**

В результате освоения дисциплины у выпускника формируются следующие компетенции:

| **Код компетенции** | **Компетенция** |
| --- | --- |
| ОК-7 | Способность к самоорганизации и самообразованию |
| ОПК-2 | Способность осваивать методики использования программных средств для решения практических задач |
| ОПК-5 | Способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности |
| ПК-1 | Способность разрабатывать модели компонентов информационных систем, включая модели баз данных и модели и интерфейсов «человек – электронно-вычислительная машина» |
| ПК-2 | Способность разрабатывать компоненты аппаратно-программных комплексов и баз данных, используя современные инструментальные средства и технологии программирования |

**Структура формирования компетенций**

В результате освоения курса «Низкоуровневое программирование» студент должен

**Знать:**

**З-1** принципы функционирования современной вычислительной системы;

**З-2** структуру современных микропроцессоров и регистровую модель микропроцессоров семейства x86/x64 и совместимых с ними;

**З-3** режимы работы и адресациимикропроцессоров семейства x86/x64 и совместимых с ними;

**З-4** систему команд языка Ассемблера для микропроцессоров семейства x86/x64 и совместимых с ними в нотациях Intel и AT&T;

**3-5** принципы программирования на Ассемблере типовых структур: ветвления, итерации, процедур, макросов, модулей;

**З-6** принципы взаимодействия программы на Ассемблере с операционной системой.

**Уметь:**

**У-1** разрабатывать алгоритмы работы низкоуровневых программ;

**У-2** разрабатывать и отлаживать программы на языке Ассемблера для микропроцессоров семейства x86/x64 и совместимых с ними;

**У-3** организовывать взаимодействие программ на языке Ассемблера как с пользователем, так и с операционной системой.

**Владеть:**

**В-1** навыками низкоуровневого программирования для операционных систем DOS/Windows и Linux;

**В-2** методами отладки низкоуровневых программ;

**В-3** принципами работы со стеком, построения многомодульных низкоуровневых программ.

|  |  |  |  |
| --- | --- | --- | --- |
| **Компетенции** | **Знания (знать)** | **Умения (уметь)** | **Навыки (владеть)** |
| ОК-7 | З-1, З-2, З-3,  З-4, З-5, З-6 | У-1, У-2, У-3 | В-1, В-2, В-3 |
| ОПК-2 | З-1, З-2, З-3,  З-4, З-5, З-6 | У-1, У-2, У-3 | В-1, В-2, В-3 |
| ОПК-5 | З-1, З-2, З-3,  З-4, З-5, З-6 | У-1, У-2, У-3 | В-1, В-2, В-3 |
| ПК-1 | З-1, З-2, З-3,  З-4, З-5, З-6 | У-1, У-2, У-3 | В-1, В-2, В-3 |
| ПК-2 | З-1, З-2, З-3,  З-4, З-5, З-6 | У-1, У-2, У-3 | В-1, В-2, В-3 |

**Программа оценивания контролируемых компетенций**

Формирование у студентов компетенций контролируется в течение всего времени освоения дисциплины в рамках:

* текущего контроля;
* рубежного контроля;
* промежуточного контроля.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **№ п.п.** | **Наименование раздела учебной дисциплины** | **Неде-ли** | **Лек-ции, час.** | **Практ. зан./ семи-нары, час.** | **Лаб. рабо-ты, час.** | **Обязат. текущий контроль (форма\*, неделя)** | **Аттестация раздела (форма\*, неделя)** | **Макси-мальный балл за раздел \*\*** | **Компетенции по разделам, проверяемые при текущем и рубежном контроле** | **Компетенции, проверяемые на зач. /экз.** |
| 4 семестр | | | | | | | | | | |
| 1 | Арифметические и логические операции. Обработка матриц. | 1-8 | 16 |  | 8 | ЛР3  ЛР6, ЛР8,  КР8 | КИ8 | 40 | ОК-7, ОПК-2, ОПК-5, ПК-1, ПК-2 |  |
| 2 | Обработка и ввод/вывод строк. Модульное программирование. Ассемблирование в Linux. | 9-15 | 14 |  | 7 | ЛР10,  ЛР12,  ЛР13  ЛР15 | КИ15 | 40 | ОК-7, ОПК-2, ОПК-5, ПК-1, ПК-2 |  |
|  | Зачет |  |  |  |  |  | З | 20 |  | ОК-7, ОПК-2, ОПК-5, ПК-1, ПК-2 |
|  | Итого за 4 семестр |  |  |  |  |  |  | 100 |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Недели** | **Содержание / Темы занятий** | **Лек., час.** | **Пр./сем., час.** | **Лаб., час.** | **Компетенции по темам, проверяемые при текущем контроле** | **Виды тек.контроля по проверке компетенций** | **Компетенции по темам, проверяемые на зач. /экз.** |
| 6 семестр | | | | | | | |
| 1 | **1. Введение.** | 2 |  | 1 |  |  |  |
| Архитектура вычислительной системы по фон Нейману. Принципы работы микропроцессоров. Регистровая модель микропроцессоров семейства х86/х64. |  |  | ОК-7, ОПК-5 |
| 2-4 | **2. Основы низкоуровневого программирования. Арифметические и логические операции.** | 6 |  | 3 |  |  |  |
| Режимы работы и адресации памяти для микропроцессоров семейства x86. Этапы разработки программы на Ассемблере. Структура программы для компилятора TASM. Регистр флагов. Арифметические команды, обработка знаковых и беззнаковых чисел. Логические команды. Команды условных и безусловного переходов. Организация циклов. Команды сдвигов. | ОК-7, ОПК-2, ОПК-5, ПК-1, ПК-2 | КР8, ЛР3, КИ8 | ОК-7, ОПК-2, ОПК-5, ПК-1, ПК-2 |
| 5-8 | **3. Работа со стеком. Обработка матриц. Процедуры и макросы в Ассемблере.** | 8 |  | 4 |  |  |  |
| Организация стека в микропроцессорах семейства x86. Команды работы со стеком. Разновидности косвенной адресации. Использование косвенной адресации для чтения данных из стека. Базовая индексная со смещением адресация как инструмент для обработки матриц. Процедуры: команды вызова и возврата. Передача параметров процедуре и коррекция стека при возврате. Макросы. | ОК-7, ОПК-2, ОПК-5, ПК-1, ПК-2 | КР8, ЛР6, ЛР8, КИ8 | ОК-7, ОПК-2, ОПК-5, ПК-1, ПК-2 |
| 9. | **4. Обработка строк.** | 2 |  | 1 |  |  |  |
| Команды обработки цепочек символов. Флаг DF. Сканирование строчек и поиск подстрок. Префиксы повторения. | ОК-7, ОПК-2, ОПК-5, ПК-1, ПК-2 | ЛР10, КИ15 | ОК-7, ОПК-2, ОПК-5, ПК-1, ПК-2 |
| 10-11. | **5. Работа с консолью и файлами.** | 4 |  | 2 |  |  |  |
| Взаимодействие с операционной системой через программные прерывания. Консоль: посимвольный ввод/вывод, ввод/вывод строк. Операции при работе с файлами: создание/открытие/закрытие, чтение/запись, удаление, смещение указателя текущей позиции. | ОК-7, ОПК-2, ОПК-5, ПК-1, ПК-2 | ЛР12, КИ15 | ОК-7, ОПК-2, ОПК-5, ПК-1, ПК-2 |
| 12 | **6. Многомодульные программы.** | 2 |  | 1 | ОК-7, ОПК-2, ОПК-5, ПК-1, ПК-2 | ЛР13, КИ15 | ОК-7, ОПК-2, ОПК-5, ПК-1, ПК-2 |
| Обобщенные директивы сегментации. Описание сегментов. Программы с несколькими сегментами кода и данных. Особенности сборки многомодульных программ. |
| 13-14 | **7. Низкоуровневое программирование в ОС семейства Linux.** | 4 |  | 2 | ОК-7, ОПК-2, ОПК-5, ПК-1, ПК-2 | ЛР15, КИ15 | ОК-7, ОПК-2, ОПК-5, ПК-1, ПК-2 |
| Структура программы на Ассемблере под Linux. Синтаксис команд Ассемблера в нотации AT&T. Системные вызовы в Linux. Обращение к функциям LIBC. |
| 15 | **8. Обработка прерываний. Резидентные программы.** | 2 |  | 1 | ОК-7, ОПК-2, ОПК-5, ПК-1, ПК-2 | КИ15 | ОК-7, ОПК-2, ОПК-5, ПК-1, ПК-2 |
| Действия микропроцессора при обработке аппаратных и программных прерываний. Обработчики прерываний. Перепрограммирование обработчиков прерываний. Резидентные программы. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Недели** | ***4 семестр***  **Лабораторные работы** | **Компетенции по темам, проверяемые при текущем контроле** | **Виды тек. контроля по проверке компетенций** | **Компетенции по темам, проверяемые на зач. /экз.** |
| 1-8 | **Раздел 1. Арифметические и логические операции. Обработка матриц.** | | | |
| 1. Арифметические операции. | ОК-7, ОПК-2, ОПК-5, ПК-1, ПК-2 | ЛР3 | ОК-7, ОПК-2, ОПК-5, ПК-1, ПК-2 |
| 2. Логические операции | ОК-7, ОПК-2, ОПК-5, ПК-1, ПК-2 | ЛР6 | ОК-7, ОПК-2, ОПК-5, ПК-1, ПК-2 |
| 3. Обработка матриц | ОК-7, ОПК-2, ОПК-5, ПК-1, ПК-2 | ЛР8 | ОК-7, ОПК-2, ОПК-5, ПК-1, ПК-2 |
| 9-15 | **Раздел 2. Обработка и ввод/вывод строк. Модульное программирование. Ассемблирование в Linux.** | | | |
| 4. Обработка строк | ОК-7, ОПК-2, ОПК-5, ПК-1, ПК-2 | ЛР10 | ОК-7, ОПК-2, ОПК-5, ПК-1, ПК-2 |
| 5. Ввод/вывод строк | ОК-7, ОПК-2, ОПК-5, ПК-1, ПК-2 | ЛР12 | ОК-7, ОПК-2, ОПК-5, ПК-1, ПК-2 |
|  | 6. Многомодульное программирование | ОК-7, ОПК-2, ОПК-5, ПК-1, ПК-2 | ЛР13 | ОК-7, ОПК-2, ОПК-5, ПК-1, ПК-2 |
|  | 7. Ассемблирование в ОС семейства Linux | ОК-7, ОПК-2, ОПК-5, ПК-1, ПК-2 | ЛР15 | ОК-7, ОПК-2, ОПК-5, ПК-1, ПК-2 |

**Соответствие оценочных средств видам контроля**

|  |  |
| --- | --- |
| **Вид контроля\*** | **Наименование оценочного средства (способ оценки: устно/ письменно /комп. технолог.)** |
| ЛР | Отчет по лабораторной работе (в электронном виде) |
| Защита лабораторной работы (устно) |
| КР | Контрольная работа (письменно) |
| КИ | Контроль по итогам выполнения (интегральная оценка без проведения дополнительного контроля) |
| З | Вопросы к зачету (устно) |

*\* Примечание*: цифра в конце обозначает номер учебной недели, на которой проводится контроль

**Структура оценки по видам и срокам контроля**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№п/п** | **Вид контроля\*** | **Максимальный балл** | **Минимальный балл** | **Примечания** |
| 1 | ЛР3, ЛР6, ЛР8, ЛР10, ЛР12, ЛР13, ЛР15 | 10 | 6 | Баллы уменьшаются на 1 с каждой неделей просрочки до достижения минимального значения |
| 2 | КР8 | 10 | 6 | При недоборе баллов до минимального значения подлежит переписыванию |
| 3 | КИ8 | 40 | 24 | Суммируются баллы за ЛР3, ЛР6, ЛР8, КР8 |
| 4 | КИ15 | 40 | 24 | Суммируются баллы за ЛР10, ЛР12, ЛР13, ЛР15 |
| 5 | З | 20 | 12 | Проводится на зачетной неделе. |

*\* Примечание*: цифра в конце обозначает номер учебной недели, на которой проводится контроль

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего профессионального образования

«Национальный исследовательский ядерный университет «МИФИ»

ФАКУЛЬТЕТ КИБЕРНЕТИКИ И ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

КАФЕДРА КОМПЬЮТЕРНЫХ СИСТЕМ И ТЕХНОЛОГИЙ (№ 12)

**Комплект материалов для оценивания выполнения лабораторных работ по дисциплине**

**«НИЗКОУРОВНЕВОЕ ПРОГРАММИРОВАНИЕ»**

|  |  |  |
| --- | --- | --- |
| Составитель | Васильев Н.П. | Доцент, к.т.н. |
| Учебный год | 2014/2015 |  |

**Список лабораторных работ, подготовка и план их выполнения**

**Лабораторная работа 1 (*ЛР3*). Арифметические операции**

**ПОДГОТОВКА К ВЫПОЛНЕНИЮ РАБОТЫ**

1. Изучить регистры микропроцессора семейства х86.
2. Изучить режимы прямой, непосредственной, неявной и косвенной адресации.
3. Изучить инструкции сложения, вычитания, умножения и деления.
4. Изучить работу компилятора TASM.EXE, редактора связей TLINK.EXE, ОТЛАДЧИКА TD.EXE
5. Разработать программу на Ассемблере для компилятора TASM согласно варианту задания.
6. Выполнить тестирование и отладку разработанной программы.

**ПЛАН ВЫПОЛНЕНИЯ РАБОТЫ**

1. Продемонстрировать работу разработанной программы преподавателю на исходных данных (значениях операндов), которые задает преподаватель.
2. Выполнить контрольное (дополнительное) задание, т.е. модифицировать разработанную программу в соответствии с дополнительными требованиями преподавателя, провести ее тестирование и отладку и продемонстрировать работу программы преподавателю.

**Контрольное ЗАДАНИЕ**

1. Изменить размер операндов в соответствии с указаниями преподавателя.
2. Изменить расчетную формулу согласно требованиям преподавателя (ввод дополнительного множителя и/или возведение в степень одного из операндов)

**ВАРИАНТЫ ЗАДАНИЙ**

1. Вычислить , операнды – слова, вычисления беззнаковые.
2. Вычислить , операнды – слова, вычисления беззнаковые.
3. Вычислить , операнды – слова, вычисления знаковые.
4. Вычислить , операнды – байты, вычисления беззнаковые.
5. Вычислить , операнды – слова, вычисления беззнаковые.
6. Вычислить , операнды – байты, вычисления знаковые.
7. Вычислить , операнды – байты, вычисления беззнаковые.
8. Вычислить , операнды – слова, вычисления знаковые.
9. Вычислить , операнды – байты, вычисления знаковые.
10. Вычислить , операнды – байты, вычисления знаковые.
11. Вычислить , операнды – слова, вычисления знаковые.
12. Вычислить , операнды – байты, вычисления знаковые.
13. Вычислить , операнды – байты, вычисления знаковые.
14. Вычислить , операнды – байты, вычисления знаковые.
15. Вычислить , операнды – байты, вычисления знаковые.
16. Вычислить , операнды – байты, вычисления беззнаковые.
17. Вычислить , операнды – байты, вычисления знаковые.
18. Вычислить , операнды – слова, вычисления знаковые.
19. Вычислить , операнды – байты, вычисления знаковые.
20. Вычислить , операнды – байты, вычисления беззнаковые.
21. Вычислить , операнды – байты, вычисления беззнаковые.

**Лабораторная работа 2 (*ЛР6*). Логические операции**

**ПОДГОТОВКА К ВЫПОЛНЕНИЮ РАБОТЫ**

1. Изучить флаги, логические инструкции, инструкции цикла, сдвигов, условных и безусловного переходов.
2. Разработать программу на Ассемблере для компилятора TASM согласно варианту задания.
3. Выполнить тестирование и отладку разработанной программы.

**ПЛАН ВЫПОЛНЕНИЯ РАБОТЫ**

1. Продемонстрировать работу разработанной программы преподавателю на исходных данных (значениях операндов), которые задает преподаватель.
2. Выполнить контрольное (дополнительное) задание, т.е. модифицировать разработанную программу в соответствии с дополнительными требованиями преподавателя, провести ее тестирование и отладку и продемонстрировать работу программы преподавателю.

**Контрольное ЗАДАНИЕ**

1. Изменить размер операндов в соответствии с указаниями преподавателя.
2. Изменить задание (требования к программе) согласно указаниям преподавателя.

**ВАРИАНТЫ ЗАДАНИЙ**

1. Определить, сколько раз в заданном слове встречается комбинация 010.

2. Заменить в заданном слове комбинации 101 на комбинации 010.

3. Построить зеркальное отображение заданного байта.

4. Задан байт. Построить слово, младший байт которого содержит исходный, а старший – его зеркальное отображение.

5. Подсчитать число единиц в заданном слове.

6. Заменить в заданном слове комбинации 1011 на комбинации 1101.

7. Подсчитать, сколько раз в заданном слове встречается комбинация 0110.

8. Задано двойное слово. Подсчитать, сколько байт в нем являются палиндромами.

9. Задан байт. Представить его в обратном коде.

10. Представить заданный байт в виде слова в прямом коде (старший байт – только знаковый).

11. Определить сумму числа нулей и единиц в заданном слове.

12. Даны два слова. Определить, сколько раз встречается комбинация 000 в остатке от деления первого слова на второе.

13. Реализовать алгоритм деления целых чисел. Делимое – слово, делитель, частное и остаток – байты.

14. Определить наибольший общий делитель двух заданных байт и построить его зеркальное отображение.

15. Определить наименьшее общее кратное двух заданных слов и представить его в обратном коде.

16. Подсчитать количество нулей в четных битах заданного двойного слова

17. Вычислить , где a и b – количество нулей и единиц заданного двойного слова соответственно

**Лабораторная работа 3 (*ЛР8*). Обработка матриц**

**ПОДГОТОВКА К ВЫПОЛНЕНИЮ РАБОТЫ**

1. Изучить режимы адресации: регистровую, базовую индексную и базовую индексную со смещением.
2. Разработать программу на Ассемблере для компилятора TASM согласно варианту задания.
3. Выполнить тестирование и отладку разработанной программы.

**ПЛАН ВЫПОЛНЕНИЯ РАБОТЫ**

1. Продемонстрировать работу разработанной программы преподавателю на исходных данных (значениях операндов), которые задает преподаватель.
2. Выполнить контрольное (дополнительное) задание, т.е. модифицировать разработанную программу в соответствии с дополнительными требованиями преподавателя, провести ее тестирование и отладку и продемонстрировать работу программы преподавателю.

**Контрольное ЗАДАНИЕ**

1. Изменить размер операндов в соответствии с указаниями преподавателя.
2. Изменить задание (требования к программе) согласно указаниям преподавателя.

**ВАРИАНТЫ ЗАДАНИЙ**

1. Дана матрица M\*N. Найти строку и столбец, содержащие минимальное количество нулевых элементов.
2. Дана матрица N\*N. Проверить ее на симметричность относительно главной и побочной диагонали.
3. Дана матрица M\*N. Поменять местами столбцы, содержащие минимальный и максимальный элементы.
4. Дана матрица N\*N. Возвести ее в заданную степень K.
5. Дана матрица M\*N. Заполнить ее с помощью линейного конгруэнтного генератора псевдослучайных чисел с заданными параметрами.
6. Дана матрица N\*N. Поменять местами строчки, симметричные друг другу относительно главной диагонали.
7. Заполнить разреженную матрицу размерностью M\*N на основе групп из трех чисел: первого индекса, второго индекса и значения элемента. Оставшиеся элементы считаются равными нулю.
8. Дана матрица N\*N. Заполнить все положительные элементы максимальным значением, а отрицательные – минимальным.
9. Дана матрица M\*N. Выделить в ней наиболее подходящую квадратную матрицу и вычислить сумму элементов ее главной диагонали.
10. Дана матрица N\*N, заполненная нулями и единицами. Поменять местами строки, содержащие минимальное и максимальное количество нулей.
11. Дана матрица M\*N. Заменить все четные элементы на минимальный, а нечетные – на максимальный.
12. Дана матрица M\*N. Повернуть ее вокруг вертикальной оси, т.е. поменять местами столбцы 1-й с последним, 2-й с предпоследним и т.д.
13. Дана матрица N\*N. Вычислить сумму диагоналей (параллельных главной), на которых расположены минимальный и максимальный элементы матрицы.
14. Дана матрица M\*N. Выполнить сортировку каждой строки методом выбора.
15. Дана матрица M\*N. Выполнить сортировку каждого столбца методом вставок.
16. Дана матрица N\*N. Выполнить сортировку элементов главной диагонали любым методом сортировки.
17. Дана матрица N\*N, изначально заполненная нулями. Заполнить ее единицами с помощью генерации индексов посредством линейного конгруэнтного генератора псевдослучайных чисел.
18. Дана матрица M\*N. Построить вектор, состоящий из сумм строк матрицы, после чего отсортировать его любым способом.
19. Дана матрица N\*N. Построить транспонированную матрицу, после чего умножить исходную на транспонированную.
20. Дана матрица M\*N. Упорядочить строки по возрастанию количества нулевых элементов в каждой из них.
21. Дана матрица M\*N. В каждом столбце поменять местами минимальный и максимальный элементы.

**Лабораторная работа 4 (*ЛР10*). Обработка строк**

**ПОДГОТОВКА К ВЫПОЛНЕНИЮ РАБОТЫ**

1. Изучить команды работы со строками (с учетом направления их обработки).
2. Изучить команды работы со стеком.
3. Разработать программу на Ассемблере для компилятора TASM согласно варианту задания.
4. Выполнить тестирование и отладку разработанной программы.

**ПЛАН ВЫПОЛНЕНИЯ РАБОТЫ**

1. Продемонстрировать работу разработанной программы преподавателю на исходных данных (значениях операндов), которые задает преподаватель.
2. Выполнить контрольное (дополнительное) задание, т.е. модифицировать разработанную программу в соответствии с дополнительными требованиями преподавателя, провести ее тестирование и отладку и продемонстрировать работу программы преподавателю.

**Контрольное ЗАДАНИЕ**

1. Изменить задание (требования к программе) согласно указаниям преподавателя.

**ВАРИАНТЫ ЗАДАНИЙ**

**Общая часть для всех вариантов:**

Дан текст, состоящий из некоторого набора строк. Каждая строка представляет собой набор слов, содержащих буквы латинского алфавита, цифры, другие печатные символы. Разделителями слов являются символы «пробел», «,», «;». Обработать каждую строку текста в соответствии с одним из указанных ниже вариантов.

1. Определить количество идущих подряд символов «1» и номера позиций, с которых начинается каждая такая серия.
2. В каждом слове строки поменять порядок символов на обратный.
3. Удалить все повторяющиеся слова в строке.
4. Отсортировать строку по возрастанию длины слов.
5. Отсортировать слова в строке по лексикографическому признаку.
6. Заменить все множественные пробелы одним, точки – символом «\*», многоточия – символом «-».
7. Выделить в каждой строке согласные буквы и отсортировать их по алфавиту.
8. Построить вещественное число, целая часть которого есть длина первого слова строки; каждая цифра в дробной части является длиной очередного слова строки.
9. Определить, сколько раз в строке повторяются одинаковые буквы.
10. Для каждой строки определить число слов разной длины.
11. Обеспечить выравнивание строк текста «по ширине» за счет равномерной вставки дополнительных пробелов между словами.
12. Зашифровать строки, используя в качестве ключа отдельно заданную строку, путем подстановки с помощью операции сложения по модулю 2. Предусмотреть операцию расшифрования.
13. Строка представляет собой арифметическое выражение, содержащее цифры, знаки «+» и «-». Вычислить результат для каждой строки.
14. Поменять порядок слов в строке на обратный.
15. Зашифровать текст посредством простого вертикального перестановочного шифра. Предусмотреть операцию расшифрования.
16. Для каждой строки построить цифровую подпись посредством суммирования всех символов по модулю 2^64. Использовать полученные сигнатуры для проверки подлинности других строк.
17. Закодировать все слова строки, поменяв местами первый и последний символы.
18. Выделить подстроки из согласных букв, справа и слева у которых расположены гласные.
19. Найти в каждой строке слова, являющиеся палиндромами.
20. Строка представляет собой подобие математической формулы. Проверить ее корректность по открывающим «(» и закрывающим «)» скобкам.

**Лабораторная работа 5 (*ЛР12*). Ввод/вывод строк**

**ПОДГОТОВКА К ВЫПОЛНЕНИЮ РАБОТЫ**

1. Изучить функции операционной системы для ввода с клавиатуры и вывода на экран
2. Изучить функции операционной системы для работы с файлами.
3. Изучить команды вызова и возврата из процедур, передачу параметров.
4. Разработать программу на Ассемблере для компилятора TASM согласно варианту задания.
5. Выполнить тестирование и отладку разработанной программы.

**ПЛАН ВЫПОЛНЕНИЯ РАБОТЫ**

1. Продемонстрировать работу разработанной программы преподавателю на исходных данных (значениях операндов), которые задает преподаватель.
2. Выполнить контрольное (дополнительное) задание, т.е. модифицировать разработанную программу в соответствии с дополнительными требованиями преподавателя, провести ее тестирование и отладку и продемонстрировать работу программы преподавателю.

**Контрольное ЗАДАНИЕ**

1. Изменить задание (требования к программе) согласно указаниям преподавателя.

**ВАРИАНТЫ ЗАДАНИЙ**

Соответствуют таковым из заданий к Лабораторной работе 4. Программа должна иметь текстовое меню, позволяющее продемонстрировать любой из режимов ее работы: ввод текста с клавиатуры/из файла, обработку текста в соответствии с заданием, вывод результата на экран либо в файл.

**Лабораторная работа 6 (*ЛР13*). Многомодульное программирование**

**ПОДГОТОВКА К ВЫПОЛНЕНИЮ РАБОТЫ**

1. Изучить обобщенные директивы сегментации и разновидности сегментов.
2. Изучить принципы разработки макрокоманд и передачу параметров в макросы.
3. Разработать многомодульную программу на Ассемблере для компилятора TASM согласно варианту задания.
4. Выполнить тестирование и отладку разработанной программы.

**ПЛАН ВЫПОЛНЕНИЯ РАБОТЫ**

1. Продемонстрировать работу разработанной программы преподавателю на исходных данных (значениях операндов), которые задает преподаватель.
2. Выполнить контрольное (дополнительное) задание, т.е. модифицировать разработанную программу в соответствии с дополнительными требованиями преподавателя, провести ее тестирование и отладку и продемонстрировать работу программы преподавателю.

**Контрольное ЗАДАНИЕ**

1. Изменить задание (требования к программе) согласно указаниям преподавателя.
2. Создать новый модуль программы, разместив в нем выбранную на усмотрение преподавателя процедуру.

**ВАРИАНТЫ ЗАДАНИЙ**

Соответствуют таковым из заданий к Лабораторным работам 4 и 5. Программа должна иметь текстовое меню, позволяющее продемонстрировать любой из режимов ее работы: ввод текста с клавиатуры/из файла, обработку текста в соответствии с заданием, вывод результата на экран либо в файл.

**Лабораторная работа 7 (*ЛР15*). Ассемблирование в ОС семейства Linux**

**ПОДГОТОВКА К ВЫПОЛНЕНИЮ РАБОТЫ**

1. Изучить синтаксис AT&T для оформления ассемблерных инструкций в Linux.
2. Изучить функции операционной системы семейства Linux для ввода с клавиатуры и вывода на экран (как системные вызовы, так и функции LIBC)
3. Изучить команды вызова и возврата из процедур, передачу параметров.
4. Разработать программу на Ассемблере для компилятора **gcc** согласно варианту задания.
5. Выполнить тестирование и отладку разработанной программы.

**ПЛАН ВЫПОЛНЕНИЯ РАБОТЫ**

1. Продемонстрировать работу разработанной программы преподавателю на исходных данных (значениях операндов), которые задает преподаватель.
2. Выполнить контрольное (дополнительное) задание, т.е. модифицировать разработанную программу в соответствии с дополнительными требованиями преподавателя, провести ее тестирование и отладку и продемонстрировать работу программы преподавателю.

**Контрольное ЗАДАНИЕ**

1. Изменить задание (требования к программе) согласно указаниям преподавателя.

**ВАРИАНТЫ ЗАДАНИЙ**

Соответствуют таковым из заданий к Лабораторным работам 4/5/6. Программа должна работать в режиме терминала ОС Linux, обеспечивая ввод исходного текста с клавиатуры, обработку его в соответствии с заданием и вывод результата на экран.

**Методика оценки результатов выполнения**

лабораторных работ по курсу «Низкоуровневое программирование» за 4 семестр

Все программы для каждой из лабораторных работ не должны иметь синтаксических ошибок, а также правильно функционировать (в соответствии с заданием) на вводимых наборах данных, как предложенных студентом, так и задаваемых преподавателем.

После демонстрации правильной работы программы студент получает контрольное (дополнительное) задание, заключающееся в незначительном изменении исходных требований к программе (условии задачи). Данное дополнительное задание должно быть выполнено строго в аудиторные часы, предусмотренные учебным планом для лабораторных занятий. Если студент не успевает сделать контрольное задание до конца отведенного времени, работа не зачитывается, и в следующий раз может быть выдано другое доп. задание.

Листинг разработанной программы, включая все контрольные задания, является отчетом по соответствующей лабораторной работе и сдается в электронной форме преподавателю.

Максимальный балл за выполнение каждой из лабораторных работ – 10 баллов, минимальный – 6. При несдаче работы в заданный срок, баллы снижаются каждую последующую неделю на 1 вплоть до указанного минимума.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего профессионального образования

«Национальный исследовательский ядерный университет «МИФИ»

ФАКУЛЬТЕТ КИБЕРНЕТИКИ И ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

КАФЕДРА КОМПЬЮТЕРНЫХ СИСТЕМ И ТЕХНОЛОГИЙ (№ 12)

**Вопросы к зачету по дисциплине**

**«НИЗКОУРОВНЕВОЕ ПРОГРАММИРОВАНИЕ»**

|  |  |  |
| --- | --- | --- |
| Составитель | Васильев Н.П. | Доцент, к.т.н. |
| Учебный год | 2014/2015 |  |

**ВОПРОСЫ К ЗАЧЕТУ**

1. Архитектура вычислительной системы по фон Нейману.
2. Принципы работы современных микропроцессоров (МП) на примере МП семейства х86/х64.
3. Режимы работы микропроцессоров семейства х86/х64.
4. Последовательность действий при загрузке ПЭВМ, построенной на базе МП семейства х86/х64.
5. Режимы адресации памяти МП МП семейства х86/х64.
6. Система команд МП семейства х86/х64. Формат ассемблерных инструкций.
7. Структура программы на Ассемблере при использовании упрощенных директив сегментации для компилятора TASM. Виды моделей памяти.
8. Структура программы на Ассемблере при использовании полноценных директив сегментации. Взаимодействие сегментов.
9. Описание данных в программах на Ассемблере для МП семейства х86/х64.
10. Инструкции пересылки, сложения, вычитания, сравнения для МП семейства х86/х64.
11. Инструкции умножения и деления для МП семейства х86/х64. Особенность обработки знаковых и беззнаковых чисел.
12. Флаги МП семейства х86/х64. Регистр флагов.
13. Команды условного перехода МП семейства х86/х64.
14. Организация циклов для МП семейства х86/х64.
15. Особенности безусловного перехода в программах для МП семейства х86/х64.
16. Логические команды МП семейства х86/х64. Организация ветвлений в программах на Ассемблере.
17. Команды сдвигов для МП семейства х86/х64.
18. Обработка матриц в программах для МП семейства х86/х64.
19. Обработка строк (цепочек символов). Копирование строк.
20. Обработка строк (цепочек символов). Загрузка и выгрузка символов.
21. Обработка строк (цепочек символов). Сканирование и сравнение строк.
22. Стек в МП семейства х86/х64. Команды работы со стеком.
23. Стек в МП семейства х86/х64. Извлечение данных из стека посредством косвенной адресации.
24. Процедуры в МП семейства х86/х64. Передача параметров и коррекция указателя стека при возврате из процедуры.
25. Макроопределения и макрорасширения в языке Ассемблера для МП семейства х86/х64.
26. Работа с консолью. Посимвольный и построчный ввод/вывод с помощью системного вызова int 21h
27. Работа с файлами. Формат имени файла. Создание, открытие, закрытие и удаление файлов с помощью системного вызова int 21h
28. Работа с файлами. Файловые дескрипторы. Чтение, запись, перемещение указателя в файлах с помощью системного вызова int 21h
29. Модульное программирование. Описание сегментов. Выравнивание, совмещение и классы сегментов. Обмен данными между модулями.
30. Особенности ассемблирования в ОС семейства Linux. Формат программы. Компиляция и отладка программ в Linux.
31. Особенности ассемблирования в ОС семейства Linux. Синтаксис команд Ассемблера в нотации AT&T
32. Особенности ассемблирования в ОС семейства Linux для 32-х и 64-х разрядных ОС. Расширенная регистровая модель МП семейства х64.
33. Обработка прерываний в ВС на базе МП семейства х86/х64. Контроллер прерываний.
34. Обработка прерываний в ВС на базе МП семейства х86/х64. Программные и аппаратные прерывания.
35. Обработка прерываний в ВС на базе МП семейства х86/х64. Перепрограммирование обработчиков прерываний. Резидентные программы.

**Методика оценки результатов сдачи зачета**

по курсу «Низкоуровневое программирование» за 4 семестр

Критерии оценки знаний устанавливаются в соответствии с требованиями к профессиональной подготовке, исходя из действующих учебных планов и программ, с учётом характера будущей практической деятельности выпускника.

Зачет проводится в устной форме, задается два вопроса из приведенного выше перечня. Максимальная оценка за зачет составляет 20 баллов, минимальная – 12 баллов.

**Итоговая оценка по курсу выставляется в соответствии**

**со следующей таблицей:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Сумма баллов по дисциплине** | **Зачет** | **Оценка (ECTS)** | **Градация** |
| 90 - 100 | Зачтено | А | Отлично |
| 85 - 89 | В | Очень хорошо |
| 75 - 84 | С | Хорошо |
| 70 - 74 | D | Удовлетворительно |
| 65 - 69 |
| 60 - 64 | E | Посредственно |
| Ниже 60 | Не зачтено | F | Неудовлетворительно |